



USAISEC

US Army Information Systems Engineering Command
Fort Huachuca, AZ 85613-5300

AD-A268 236



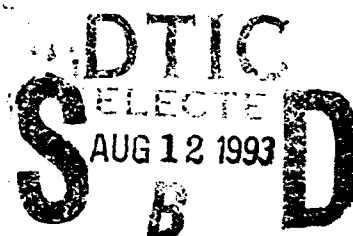
2

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
COMMUNICATIONS, AND COMPUTER SCIENCES

The Software Support Qualitative Assessment Methodology

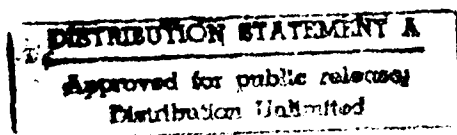
Volume III

Implementing the Software Supportability Measure



March 1991

ASQB-GI-91-018



93-18409



AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800

93



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704--188
Exp. Date: Jun 30, 1996

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE	
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION / AVAILABILITY OF REPORT N/A	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ASQB-GI-91-018			5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A	
6a. NAME OF PERFORMING ORGANIZATION The Center for Information Management Research		6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION N/A	
6c. ADDRESS (City, State, and ZIP Code) SERC Georgia Institute of Technology Atlanta, GA 30332		Dept of MIS University of Arizona Tucson, AZ 85613	7b. ADDRESS (City, State, and Zip Code) N/A	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AIRMICS		8b. OFFICE SYMBOL (if applicable) ASQB - GI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) 115 O'Keefe Bldg., Georgia Institute of Technology Atlanta, GA 30332-0800		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO. 62783A	PROJECT NO. DY10	TASK NO. 02-01-01
				WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) The Software Support Qualitative Assessment Methodology Volume III (UNCLASSIFIED) Implementing the Software Supportability Measure				
12. PERSONAL AUTHOR(S) W. Michael McCracken, Elizabeth Mynatt, Christopher Smith (GIT) J.F. Nunamaker, Ai-Mei Chang, Titus Purdin, Richard Orwig, Amit Vyas (Univ of Arizona)				
13a. TYPE OF REPORT final report		13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1991, March, 22	15. PAGE COUNT 56
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) life cycle metrics; systems support; support measures; qualitative assessment; supportability measures; assessment measures; readiness measures; information systems; software maintenance; support tools; tools management;	
FIELD				
GROUP				
SUB-GROUP				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The Software Supportability Qualitative Assessment Methodology is a five volume reference set that provides measures to aid in the support of information systems. The volumes are aimed at improving the support process by more accurately assessing the capabilities of support organizations, qualitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems. The five volumes are: I. Developing Quality Measures for Information Systems Support II. The Review of Metrics for Developing an Information Systems Support Measurement Framework III. Implementing the Software Supportability Measure IV. Implementing the Support Organization Assessment Measure V. Implementing the Operational Readiness Measure This volume provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the supportability of an information system based on its evaluation. Specifically, information is presented that contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the supportability of an information system.				
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED / UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Howard C "Butch" Higley			22b. TELEPHONE (Include Area Code) (404) 894-3110	22c. OFFICE SYMBOL ASQB-GI

The research herein was performed for the Army Institute for Research in Management Information, Communications, and Computer Sciences (AIRMICS), the RDTE organization of the U.S. Army Information Systems Engineering Command (USAISEC). The sponsor for the project was the Office of the Director of Information Systems for Command, Control, Communications, and Computers (ODISC4). The principal investigator was from The Center for Information Management Research (CIMR), W. Michael McCracken of the Georgia Institute of Technology.

This research report is not to be construed as an official Army position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited, and is not protected by copyright laws. Your comments on all aspects of the document are solicited.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 3

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED

s/ _____
 Glenn E. Racine
 Chief
 CISD

s/ _____
 John R. Mitchell
 Director
 AIRMICS

**The Software Support
Qualitative Assessment Methodology
Volume III
Implementing the Software Supportability Measure**

Prepared by
The Center for Information Management Research
for the
U.S. Army Institute for Research in Management
Information, Communications, and Computer Science
(AIRMICS)
Contract No. ECD-8904815

W. Michael McCracken, Elizabeth Mynatt, Christopher Smith
Software Engineering Research Center
Georgia Institute of Technology

December 1990

The **Software Supportability Qualitative Assessment Methodology** is a five volume reference set that provides measures to aid in the support of information systems. These manuals are aimed at improving the support process by more accurately assessing the capabilities of support organizations, quantitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems.

Volume I, *Developing Quality Measures for Information Systems Support*, describes the three measures along with the model of information system support that the measures are designed to satisfy. This is the main volume of the set and should be consulted before implementing the measures described in more detail in the other volumes.

Volume II, *The Review of Metrics for Developing an Information Systems Support Measurement Framework*, provides a survey and evaluation of current metrics in terms of information systems support. Specifically, three classes of metrics are reviewed: software product metrics, life cycle process metrics, and process management metrics.

Volume III, *Implementing the Software Supportability Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the supportability of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the supportability of an information system.

Volume IV, *Implementing the Support Organization Assessment Measure*, provides instructions for collecting data for the assessment, conducting the assessment, and interpreting the final result. The volume also contains guidelines for improving the capabilities of a support organization based on its evaluation. Specifically, the volume contains resource estimations for conducting and evaluating the assessment, questionnaires for collecting the required data and step-by-step instructions for measuring the capabilities of a support organization.

Volume V, *Implementing the Operational Readiness Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the operational readiness of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the operational readiness of an information system.

Contents

1 Introduction	2
2 Requirements	2
Material	2
Audience	2
Time	3
3 Calculating and Evaluating the Supportability Measure	4
4 Scoring the Questionnaire Answers	5
5 Interpreting the Results	6
A Glossary of Terms	9
B List of Acronyms	12
C Organizational Questionnaire	13
D System Questionnaire	14
E Scoring Directions	15
F Supportability Worksheet - Final Results	16

1 Introduction

The software supportability measure is focused on defining the important factors that affect the supportability of a fielded information system. Software supportability is a measure of the effort required to satisfy users expectations of a given software product, where user expectations can be divided into two groups. First the users expect the software to fulfill its intended functions, i.e. its requirements. Second, users generally expect the software to meet new requirements. Factors affecting the effort required to satisfy these expectations can be divided into three categories: the software product itself, the available resources for support activities, and the management procedures used to guide the support process.

The purpose of this measure is to give the support organization a rough characterization of the supportability of an information system supported by the organization. The measure is made up three factors: *system*, *process*, and *resource*. The *system* factor measures criteria related solely to the information system. The *process* factor measures components related to the maturity and effectiveness of the process used to guide system support. The *resource* factor measures components related to the availability and effectiveness of resources critical to system support.

This document describes how to measure the supportability of an information system that is maintained by your organization. The second section details what resources in time, material, and personnel are required to compute this measure. The following sections describe the process for computing and interpreting the supportability measure.

It is important you read through the following two sections (up through Calculating and Evaluating the Supportability Measure) before beginning this process. It is also important that any personnel who will provide data for this process (by completing one or more questionnaires) do NOT read the sections on scoring the questionnaires and evaluating the results (Sections 4 and 5) until after they have completed the questionnaires.

2 Requirements

Material

Little in the way of materials is required to conduct this exercise. Appendices C and D in this volume contain the Organization and System questionnaire. These should be photocopied and distributed to the appropriate personnel (see next subsection). Appendix E contains directions and worksheets for scoring the questionnaires. A set of directions and a worksheet should be photocopied for each questionnaire. Appendix F contains a worksheet that should be used to record the final results of the supportability measurement. This worksheet should be photocopied and should be used in interpreting and evaluating the final measurement.

Audience

The careful selection of the appropriate personnel to complete the organization and system questionnaires is critical to the success of the measure. An examination of the questionnaires should be the first step in choosing your audience. In general, the organization questionnaire should be

completed by personnel tasked with managing the support process. Likewise the system questionnaire should be completed by personnel tasked with the actual maintenance of the system.

Recognizing that a significant portion of the questionnaires ask for subjective answers, distributing the questionnaires to a set of respondents and averaging their scores should reduce bias accompanying subjective responses.

Selecting a coordinator to distribute, collect, validate, and score the questionnaires is required. The coordinator is responsible for distributing the questionnaires and answering any remaining questions the respondents may have. The coordinator must also collect the questionnaires, verifying that all questions have been answered completely. The coordinator must also validate the questionnaires against each other. Essentially the coordinator assures that the answers make sense (i.e. percentages add up to 100) and that the respondents interpreted the questions in the same manner. More information on this process can be found in the next section. Finally the coordinator is the best person to be responsible for scoring the questionnaire and compiling the final results. The coordinator may NOT complete any of the questionnaires as a respondent.

In summary, this process requires a minimum of three personnel to answer the organization and system questionnaire and to serve as coordinator, respectively. The final results will be more meaningful if two sets of people (possibly overlapping) complete the two questionnaires.

Time

The amount of time required to conduct the measurement depends on two factors: the amount of on-line or easily accessible data and the number of personnel tasked to complete questionnaires.

The organizational questionnaire should take from 4 person-hours to 24 person-hours to complete. The size of the organization and the amount of easily accessible information about the staff determines this variation.

The system questionnaire should take from 4 person-hours to 12 person-hours to complete. Again, the amount of readily accessible information determines this variation.

Each questionnaire should take one half of a person-hour to score.

The amount of time required of the coordinator is determined by the number of personnel filling out questionnaires. Validating the questionnaires should take approximately one person-hour per questionnaire. The effort should be less for a small number of questionnaires.

A rough formula to calculate the time required for collecting the data and calculating the measure is given on the next page.

OW = Weight based on size of organization and accessibility of information for the staff. Range from 1 (small organization and readily accessible information) to 6 (large organization etc..).

SW = Weight based accessibility of information for the system.
Range from 1 (readily accessible information) to 3.

ON = Number of organizational questionnaires completed.

SN = Number of systems questionnaires completed.

TT = Total time required in person-hours.

$$TT = 4*OW*ON + 4*SW*SN + (ON + SN) + .5*(ON + SN)$$

			- scoring
			- validation
			- completing system questionnaires
			- completing organizational questionnaires

3 Calculating and Evaluating the Supportability Measure

This process consists of six steps.

1. Select respondents and coordinator.
2. Review questionnaires (optional).
3. Fill out questionnaires.
4. Validate questionnaires.
5. Score questionnaires, compute measure.
6. Interpret final result.

First, the personnel tasked with completing the questionnaires and the overall coordinator should be selected. Refer to the earlier section for guidelines for selecting questionnaire respondents. It is possible, even desirable, for the two sets of respondents to overlap. In other words,

some people could fill out both system and organization questionnaires. This overlap enables different perspectives to be incorporated. A portion of the organizational questionnaire (questions 5 - 8) concerns educational and other historical information about the application staff. This information may be inaccessible to some respondents. If necessary, this information could be supplied by other person, perhaps the coordinator, since this information is strictly quantitative and not subjective.

The system questionnaire included in this volume is the same questionnaire that is used to compile the operational readiness measure for an information system (see Volume V). **To compute the supportability measure only the starred (*) questions (questions 1-10,14,15,19,26a-j,26o) need to be completed.** If you are computing both the supportability and operational readiness measures, the entire system questionnaire needs to be completed.

Optionally, after the respondents have been selected, a meeting could be held to review the questionnaires. This meeting should be led by the coordinator. The purpose of this meeting is to assure the respondents understand the questions in the same manner. Discussions about possible answers should not be permitted. Only definitional information should be distributed. The advantage of this meeting is that it should quicken the completion of the questionnaires by the respondents and it should reduce the variability in their interpretation of the questions.

Next, the questionnaires should be completed. The coordinator should be available to answer questions of interpretation. Respondents should be encouraged to write comments concerning interpretation next to their answers. This effort will aid the coordinator in validating the questionnaires.

The questionnaires should be returned to the coordinator who should attempt to validate the responses. First, all questions should be answered! Second, responses containing quantitative, non-subjective data should correspond closely if not be equivalent. Third, the coordinator should look for differing interpretations by examining comments added by the respondents.

Next, the coordinator should score the questionnaires. Refer to the next section and appendices E and F for scoring directions. The system questionnaire scoring directions contains only directions for scoring the questions needed for the supportability measure. These directions should be used for computing the supportability measure. Scoring directions for computing the operational readiness measure can be found in Volume V.

The coordinator should average the scores to compute a final score. Finally, the coordinator and other personnel should consult Section 5 in order to interpret the final result.

4 Scoring the Questionnaire Answers

Appendices E and F contain directions for scoring the individual questionnaires and computing the final measure. Essentially, each answer will correspond to a certain number of points. Scoring directions for each question are provided with possible ranges specified. The results for each questionnaire are then recorded on the questionnaire worksheets. The worksheets divide the responses among three categories: System, Process, and Resource. These categories are the three major factors of the supportability measure. The columns for each category should be totaled. The totals for the organizational questionnaires should then be averaged by the number of organizational questionnaires completed; likewise for the system questionnaires. These averages should be

recorded on the Supportability Worksheet. Instructions on the Supportability Worksheet should then be followed to compute the final result.

5 Interpreting the Results

Software supportability is a measure of the effort required to satisfy users expectations of a given software product, where user expectations can be divided into two groups. First the users expect the software to fulfill its intended functions, i.e. its requirements. Second, users generally expect the software to meet new requirements. Factors affecting the effort required to satisfy these expectations can be divided into three categories: the software product itself, the available resources for support activities, and the management procedures used to guide the support process.

The Supportability Measure

The supportability measure for an information system can be interpreted in two ways. First, it can be interpreted as the percentage chance of successfully fulfilling user expectations. This interpretation is rather severe, though. The following ranges may be more meaningful:

76-100 Excellent rating, low risk system

51-75 Good rating, average risk system

26-50 Mediocre rating, requires attention to reduce risk

7-25 Poor rating, requires immediate attention

The final measurements for each factor of the supportability measure are also meaningful. The *system* factor measures components related solely to the information system. The *process* factor measures components related to the maturity and effectiveness of the process used to guide system support. The *resource* factor measures components related to the availability and effectiveness of resources critical to system support.

The System Factor

Criteria for the system factor considers the existence and quality of system documentation, the age and change history of the system, the complexity, size and modularity of the source code, the existence of debugging code and adequate testing, the urgency of change requests, and the quality of the original system. The final score for the system factor can range from 2 to 100. The following interpretation may be useful:

76-100 Excellent rating, low risk system

51-75 Good rating, average risk system

26-50 Mediocre rating, implement improvements

2-25 Poor rating, system not supportable

Depending on the severity of the rating, the following actions may be taken to improve the system factor. This list does not represent all possible appropriate actions.

- Re-design the system (reverse engineering).
- Create/update system documentation.
- Increase testing time.
- Accept fewer emergency change packages.
- Consistently use regression testing.

The Process Factor

Criteria for the process factor considers the existence and adequacy of effective organizational techniques, the existence and adequacy of important standards, the existence of useful work methods, training of the user population, and adequate forecasting of resource requirements. The final score for the process factor can range from 0 to 100. The following interpretation may be useful:

76-100 Excellent rating; mature, effective process

51-75 Good rating, average process

26-50 Mediocre rating, implement improvements

0-25 Poor rating, process requires immediate attention

Process improvements are difficult to make. Examining questionnaire responses may provide clearer direction. Depending on the severity of the rating, the following actions may be taken to improve the process factor. This list does not represent all possible appropriate actions.

- Incorporate known effective organizational techniques such as maintenance escort, acceptance review, or object-oriented design.
- Create / update needed standards.
- More strictly enforce standards.
- Increase user training.
- Track resource utilization against resource estimations.

The Resource Factor

Criteria for the resource factor considers the training, experience, and morale of the application staff, budget constraints, existence of adequate, up-to-date software engineering tools, competing demands placed upon the application staff, the adequacy of existing hardware/software configurations, and the availability of qualified personnel. The final score for the resource factor can range from 0 to 100. The following interpretation may be useful:

76-100 Excellent rating, sufficient resource utilization

51-75 Good rating, average resource efficiency and availability

26-50 Mediocre rating, implement improvements

5-25 Poor rating, inadequate resources

Depending on the severity of the rating, the following actions may be taken to improve the resource factor. This list does not represent all possible appropriate actions.

- Increase programmer training.
- Increase morale, reward maintenance efforts over development efforts.
- Evaluate/install new software engineering tools.
- Increase recruiting efforts to obtain qualified programmers.
- Evaluate existing hardware/software configurations against the existing portfolio of information systems supported by the organization.
- Minimize the number of languages and work practices used for implementation in order to reduce training requirements.

A Glossary of Terms

Acceptance Review A review of a software product by developers and maintainers to determine if the product satisfies all originally specified requirements.

Acceptance Test Testing led by the client or QA group to determine whether the product satisfies its specifications as claimed by the developer.[Sch90]

Application System same as Information System

Availability A measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at a random point in time.[Dep82]

Benchmark Testing Evaluation of the system performance against quantitative requirements.[Sch90]

Change Request Review Board An authority responsible for evaluating and approving requests for changes to a software product.

Cohesion A measure of the degree of the functional relatedness within program units. [Som89]

Complexity A characteristic of the software interface which influences the resources another system will expend or commit while interfacing with the software. [CDS86]

Configuration Management The process of identifying and defining the configuration items (hardware/software units) in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items.[IEE83]

Consistency The extent to which uniform design techniques and notation are used. [War87]

Coupling A measure of the strength of interconnections (dependencies) between program units. [Som89]

Error Human action that results in software containing a fault. Examples include omission or misinterpretation of user requirements in a software specification, incorrect translation or omission of a requirement in the design specification. [IEE83]

Failure A departure of program operation from program requirements.[IEE83]

Failure Rate The number of failures of an item per measure-of-life unit.[Dep82]

Fault A manifestation of an error in software. A fault, if encountered, may cause a failure. Synonymous with bug.

Fourth Generation Language (4GL) A computer programming language that provides abstractions of data and/or procedural specifications and is usually suited for a particular application domain.

Integration Testing Verify that the modules of the system combine correctly in order to achieve a product that meets its specifications. [Sch90]

IS (Information Systems) Organization An organized collection of procedures, personnel, and resources dedicated to support a portfolio of information systems.

Lines of Code Lines of source code, not including comments.

Maintainability The probability that an item will be retained in, or restored to, a specified condition within a given period if prescribed procedures and resources are used.[Dep82]

Maintenance All actions required to retain an item in, or restore it to, a specified condition.[Dep82]

Maintenance Audit An organized review of the maintenance organization.

Maintenance Escort Participation of the software maintainer in software system development.

Man/Machine Interface The software that supports the interaction between the user and the system.

Measure A high-level unit of specification which characterizes, evaluates, or predicts various aspects of software life cycle processes and products.

Metric A measurable indication of some aspect of a system. [DeM82] A quantification of a specific feature of the software life cycle process or software product.

Modularity A characteristic of software such that it is well-structured, highly cohesive, and minimally coupled. [War87]

New Systems Development The development of a system which has never been fielded.

Object Oriented Design Designing a system in terms of abstract data types where the objects are instantiations of the data types and new data types can be defined as extensions of previously defined types.

Regression Testing Testing the system against previous test cases to ensure that the functionality of the system has not been compromised by recent changes to the system. [Sch90]

Reliability The probability that an item will perform its intended function for a specified interval under stated conditions.[Dep82]

Self-Descriptiveness A characteristic of software that enables the understanding of implementation of software functions. [War87]

Support Staff The personnel tasked with maintaining an information system.

Supportability A measure of the adequacy of products, resources, and procedures to facilitate the support activities of modifying and installing software, establishing an operational software baseline, and meeting user requirements. [PTH87]

Testability The extent to which software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria. [IEE83]

Throw-away prototyping Creating a prototype as part of system design and then "throwing away" the prototype and implementing the system "from scratch" not using any of the source code from the prototype.

Top-down design Designing the system by recursively breaking the system down into smaller components.

Unit Testing Testing of individual portions of the system.

B List of Acronyms

AIRMICS U.S. Army Institute for Research in Management Information, Communications,
and Computer Science

AMC Army Materiel Command

CCB Change Control Board

COE Army Corps of Engineers

FORSCOM Forces Command

HSC Army Health Services Command

IS Information System

ISC Army Information Systems Command

LOC Lines of Code

C Organizational Questionnaire

This appendix contains a 5 page questionnaire for gathering information about the support organization in order to calculate the supportability measure. The questionnaire should be photocopied and distributed to selected respondents.

SSQAM -- Supportability Measure
Organization Questionnaire

QUESTIONNAIRE NUMBER _____

1. Which of the following organizational techniques are established by the IS organization for application system maintenance? Indicate if these techniques are adequate.

Exist (yes or no)		
		Adequate (yes or no)
		Maintenance escort (participation of maintainer in system development.)
		Acceptance review (in transferring software from development to maintenance.)
		Change request review board (i.e. CCB)
		Formal retest procedure (in implementing changes)
		Scheduled maintenance (changes batched and implemented according to predetermined schedule.)
		Quality assurance

2. Do you have a set of standards to follow when performing the following actions? Are these standards adequate?

Exist (yes or no)		
		Adequate (yes or no)
		Developing or modifying requirements documentation
		Developing or modifying design documentation
		Developing or modifying the man/machine interface
		Developing or modifying system documentation
		Developing or modifying source code
		Conducting the software unit tests
		Conducting the software integration tests
		Conducting the software acceptance tests

SSQAM -- Supportability Measure
Organization Questionnaire

3. Are there formalized plans or procedures for conducting the following activities? (Check all those which apply.)

- ☐ Monitoring of support staff performance
- ☐ Tracking of resource utilization
- ☐ Coordination of tasks within the support staff
- ☐ Estimating resources (time, personnel...) necessary to implement changes to software
- ☐ Periodic maintenance audit
- ☐ Handling user change requests
- ☐ Designing new systems to replace existing systems
- ☐ Monitoring planned maintenance activities
- ☐ Configuration management
- ☐ Creating system test data
- ☐ Training the support staff
- ☐ Training the system users

4. Which of the following work methods are established by the IS organization for application system development and maintenance? (Check all those which apply.)

- ☐ Throw-away prototyping
- ☐ Object Oriented Design Methodology
- ☐ Top-down design
- ☐ Regression testing
- ☐ Benchmark testing

5. What is the current total number of (full-time equivalent) application systems analysts and programmer employees

total application staff

SSQAM -- Supportability Measure
Organization Questionnaire

6. What is the length of service (in the IS Organization) distribution of the current application staff? Indicate a percentage for each category. (Percentages should total to 100%)

_____ 0-1 years
_____ 1-3 years
_____ 3-6 years
_____ 6-10 years
_____ more than 10 years

7. What is the distribution of (immediate) prior job experience of the current application staff? Indicate a percentage for each category. (Percentages should total to 100%)

_____ position in other IS organization within parent organization.
_____ other position within parent organization.
_____ position in other IS organization, not in parent organization.
_____ other position, not in parent organization.
_____ no prior position (student).

8. What is the distribution of educational backgrounds (highest degrees obtained) of the application staff? Indicate a percentage for each category. (Percentages should total to 100%)

_____ Graduate college degree
_____ Bachelors college degree
_____ Two-year college degree
_____ High school diploma or less

SSQAM -- Supportability Measure
Organization Questionnaire

9. Overall, in your judgement, to what extent are (or have been) the problems in maintaining the current installed application portfolio? (Check the appropriate category.)

	No Problem At All			
	Somewhat Minor Problem			
	Minor Problem			
	Somewhat Major Problem			
	Major Problem			

a. Turnover of maintenance personnel					
b. Quality of application system documentation					
c. Changes made to application system hardware and software					
d. User demand for enhancements and extensions to application system					
e. Skills of maintenance programming personnel					
f. Quality of original programming of application system					
g. Number of maintenance programming personnel available					
h. Competing demands for maintenance programming personnel time					
i. Inadequate hardware/software configurations in IS Organization					
j. Ability to recruit qualified personnel					
k. Lack of user understanding of application system					
l. Storage requirements of application system programs					
m. Inadequate software tools (debuggers, analyzers, etc..)					
n. Motivation of maintenance programming personnel					

SSQAM -- Supportability Measure
Organization Questionnaire

	No Problem At All				
	Somewhat Minor Problem				
	Minor Problem				
	Somewhat Major Problem				
	Major Problem				

o. Forecasting of maintenance programming personnel requirements					
p. Maintenance programming productivity					
q. Competing demands between new systems development and maintenance					
z. Turnover in user organization					
s. Unrealistic user expectations					
t. Adherence to programming standards in maintenance					
u. Management support of application system					
v. Quality of application system design					
w. Budgetary pressures					
x. Meeting scheduled commitments					
y. Inadequate training of user personnel					

D System Questionnaire

This appendix contains a 7 page questionnaire for gathering information about the information system. To compute the supportability measure only the starred (*) questions (questions 1-10,14,15,19,26a-j,26o) need to be completed. If you are computing both the supportability and operational readiness measures, the entire system questionnaire needs to be completed. The questionnaire should be photocopied and distributed to selected respondents.

Software Supportability Qualitative Assessment Methodology
System Questionnaire

QUESTIONNAIRE NUMBER: _____

Name of Information System: _____

Software and Documentation Information

*1. What is the size of the system source code, in lines of code (LOC)?

_____ lines of code

*2. What language(s) is the software written in?

*3. How many modules (units that perform single functions or sets of functions) does the software product contain?

_____ number of modules

*4. What is the age (measured from date of original installation) of the software product?

_____ age of system (in years)

*5. How long has your organization supported this software product?

_____ length of support (in years)

*6. What are the TOTAL number of changes that have been made to this product (software and associated documentation) during the time you have supported it? Include both Software Change Packages and Emergency Change Packages.

_____ total number of changes

*7. Does the software contain any code that aids in debugging the software?

_____ yes
_____ no

Software Supportability Qualitative Assessment Methodology
System Questionnaire

*8. Is there any documentation explaining the overall function of the software?

_____ yes
_____ no

*9. Is there documentation for each module explaining the module's function?

_____ yes
_____ no

*10. Are there any user's manuals explaining the use of this software?

_____ yes
_____ no

Maintenance Information

11. For what amount of time (how many hours) during the month, if any, is the software system down and cannot be used?

_____ (hours) down time

12. What is the average number of maintenance requests per month received for this system?

[Notes: If a change proposal contains several requests, count each request separately.

Count ALL requests, even those that no actions are taken on.]

_____ average number of maintenance requests per month

13. Approximately how many of the above maintenance requests (per month) ultimately result in some change being made to the software?

_____ percentage of requests (per month) which result in changes to the software

*14. Approximately what percentage of the maintenance requests FOR WHICH YOU PERFORM ACTIONS ON are

_____ Small-scale (affect a few lines of code at most)?

_____ Medium-scale (affect several functions or modules)?

_____ Large-scale (affect all or a large portion of the software)?

100 % TOTAL

Software Supportability Qualitative Assessment Methodology
System Questionnaire

*15. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ EMERGENCY (require immediate attention and must be completed as
soon as possible to ensure the correct operation of the software)

_____ URGENT (require urgent attention - more so than normal requests -
and must be completed within a relatively short period of time)

_____ NORMAL (require no special attention and can be completed within
the usual framework of support procedures)

100 % TOTAL

16. What percentage of ALL maintenance requests you receive...

_____ Are for corrections to faulty software components?

_____ Are for changes (other than corrections) or enhancements to the
software?

100 % TOTAL

17. What percentage (0-100%) of EMERGENCY and URGENT requests are for
corrections to faulty software components?

_____ percentage of EMERGENCY and URGENT requests that are corrections

18. ON THE AVERAGE, what percentage (0-100%) of all requests require more time
to complete than is originally scheduled?

_____ percentage of all requests completed behind schedule

*19. What percentage of time spent maintaining the software is devoted to
testing it?

_____ (%) time spent on testing

Software Supportability Qualitative Assessment Methodology
System Questionnaire

User Information

20. ON THE AVERAGE, how often do you communicate (either formally or informally) with a TYPICAL user organization using this information system? Mark the one appropriate response below.

_____ Several times a day
_____ Once or twice a day
_____ At least weekly, but not daily
_____ At least monthly, but not weekly
_____ At least once per year, but not monthly
_____ Less than once per year

Current Circumstances

21. How many people in your support organization presently maintain this software either on a part-time or full-time basis?
(Indicate the number in each category.)

_____ full-time (number)
_____ part-time (number)

22. AT PRESENT (NOT on the average), how many changes of all types (including corrections and enhancements) are there to be implemented?

_____ number of changes to be implemented

23. Of the above changes to be implemented, what percentage (0-100%) of these changes are EMERGENCY changes? If there are no changes, answer 0%.

_____ percentage of current changes that are EMERGENCY

24. Of the changes (from #2) to be implemented, what percentage (0-100%) of these changes are for CORRECTIONS to faulty software components? If there are no changes, answer 0%.

_____ percentage of current changes that are CORRECTIONS

Software Supportability Qualitative Assessment Methodology System Questionnaire

25. Based on the following scale, how you you rate the estimated effort needed to complete changes to the software product over the next month:

- 0 = Much more effort than average
- 1 = Somewhat more effort than average
- 2 = Average effort
- 3 = Less than average effort
- 4 = Much less than average effort
- 5 = No effort at all (no changes to implement)

_____ answer (0-5)

Problem Information

*26. Overall, in your judgment, to what extent are (or have been) the following problems in maintaining this information system?
(Check the appropriate category.)

	No Problem At All				
	Somewhat Minor Problem				
	Minor Problem				
	Somewhat Major Problem				
	Major Problem				
*a. Not enough people to support this system.					
*b. People supporting this system are not trained adequately.					
*c. System is overly large, making support difficult.					
*d. System is overly complex, making support difficult.					
*e. System is not well-structured (written in "spaghetti code").					
*f. Lack of system modularization makes changes difficult to implement.					

Software Supportability Qualitative Assessment Methodology
System Questionnaire

26 (cont'd)

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem
*g. System is old and needs to be replaced.					
*h. System documentation is incomplete or confusing.					
*i. System documentation is out-of-date.					
*j. Not enough time is spent on testing after changes are made.					
k. Software repair schedules are hard to meet.					
l. Overall, there are more change requests submitted for this system than can be handled.					
m. There are too many change requests resulting from software bugs (vs. enhancement requests).					
n. There are too many emergency change requests.					
*o. User requirements for this system change frequently.					

Software Supportability Qualitative Assessment Methodology
System Questionnaire

27. Overall, from your perspective, to what extent are (or have been) the problems as they impact on the ability to maintain this information system? (Check the appropriate category.)

	No Problem At All				
	Somewhat Minor Problem				
	Minor Problem				
	Somewhat Major Problem				
	Major Problem				
a. Skills of maintenance programming personnel					
b. Number of maintenance programming personnel available					
c. Inadequate hardware/software configurations in IS Organization					
d. Motivation of maintenance programming personnel					
e. Maintenance programming productivity					
f. Competing demands between new systems development and maintenance					
g. Budgetary pressures					
h. Meeting scheduled commitments					

E Scoring Directions

This appendix contains the following four items:

1. A one-page worksheet for recording scores from the organizational questionnaire.
2. Nine pages of directions for scoring the organizational questionnaire.
3. A one-page worksheet for recording scores from the system questionnaire.
4. Ten pages of directions for scoring the system questionnaire.

SSQAM -- Supportability Measure
Organizational Questionnaire Worksheet

QUESTIONNAIRE NUMBER _____

Question	System	Process	Resource	Range
1.		_____		0-12
2.		_____		0-16
3.		_____		0-12
4.		_____		0-5
5.				
6.			_____	2-10
7.			_____	2-10
8.			_____	1-10
9a.			_____	0-5
b.	_____			0-5
c.	_____			0-5
d.			_____	0-5
e.			_____	0-5
f.	_____			0-5
g.			_____	0-5
h.			_____	0-5
i.			_____	0-5
j.			_____	0-5
k.		_____		0-5
l.			_____	0-5
m.			_____	0-5
n.		_____		0-5
o.		_____		0-5
p.			_____	0-5
q.		_____		0-5
r.			_____	0-5
s.		_____		0-5
t.		_____		0-5
u.		_____		0-5
v.	_____			0-5
w.			_____	0-5
x.		_____		0-5
y.		_____		0-5
TOTALS	_____			0-20
		_____		0-90
			_____	5-90

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

QUESTIONNAIRE NUMBER _____

1. Which of the following organizational techniques are established by the IS organization for application system maintenance? Indicate if these techniques are adequate.

Exist (yes or no)	
	Adequate (yes or no)
	Maintenance escort (participation of maintainer in system development.)
	Acceptance review (in transferring software from development to maintenance.)
	Change request review board (i.e. CCB)
	Formal retest procedure (in implementing changes)
	Scheduled maintenance (changes batched and implemented according to predetermined schedule.)
	Quality assurance

SCORING

For each affirmative answer, score 1 point.
Maximum of 12 points possible.

SCORE = _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

2. Do you have a set of standards to follow when performing the following actions? Are these standards adequate?

Exist (yes or no)		
		Adequate (yes or no)
		Developing or modifying requirements documentation
		Developing or modifying design documentation
		Developing or modifying the man/machine interface
		Developing or modifying system documentation
		Developing or modifying source code
		Conducting the software unit tests
		Conducting the software integration tests
		Conducting the software acceptance tests

SCORING

For each affirmative answer, score 1 point.
Maximum of 16 points possible.

SCORE = _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

3. Are there formalized plans or procedures for conducting the following activities? (Check all those which apply.)

- _____ Monitoring of support staff performance
- _____ Tracking of resource utilization
- _____ Coordination of tasks within the support staff
- _____ Estimating resources (time, personnel...) necessary to implement changes to software
- _____ Periodic maintenance audit
- _____ Handling user change requests
- _____ Designing new systems to replace existing systems
- _____ Monitoring planned maintenance activities
- _____ Configuration management
- _____ Creating system test data
- _____ Training the support staff
- _____ Training the system users

SCORING

For each check, score 1 point.
Maximum of 12 points possible.

SCORE - _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

4. Which of the following work methods are established by the IS organization for application system development and maintenance? (Check all those which apply.)

- _____ Throw-away prototyping
_____ Object Oriented Design Methodology
_____ Top-down design
_____ Regression testing
_____ Benchmark testing

SCORING

For each check, score 1 point.
Maximum of 5 points possible.

SCORE - _____

5. What is the current total number of (full-time equivalent) application systems analysts and programmer employees

_____ total application staff

SCORING

No scoring for this question.

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

6. What is the length of service (in the IS Organization) distribution of the current application staff? Indicate a percentage for each category. (Percentages should total to 100%)

	WEIGHT

_____ 0-1 years	1
_____ 1-3 years	2
_____ 3-6 years	3
_____ 6-10 years	4
_____ more than 10 years	5

SCORING

For each percentage, multiply by the corresponding weight. Sum the products and then divide the sum by 50. Maximum score is 10 points, minimum score is 2 points.

Example: For these percentages, the calculations are as follows:

25% 0-1 years
30% 1-3 years
20% 3-6 years
15% 6-10 years
10% more than 10 years

$$(25 * 1) + (30 * 2) + (20 * 3) + (15 * 4) + (10 * 5)$$

50

SCORE - _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

7. What is the distribution of (immediate) prior job experience of the current application staff? Indicate a percentage for each category. (Percentages should total to 100%)

	WEIGHT -----
_____ position in other IS organization within parent organization.	5
_____ other position within parent organization.	3
_____ position in other IS organization, not in parent organization.	3
_____ other position, not in parent organization.	1
_____ no prior position (student).	1

SCORING

For each percentage, multiply by the corresponding weight. Sum the products and then divide the sum by 50. Maximum score is 10 points, minimum score is 2 points.

Example: For these percentages, the calculations are as follows:

25% position in other IS organization within parent..
30% other position within parent organization
20% position in other IS organization, not in
15% other position, not in parent organization
10% no prior position (student)

$$(25 * 5) + (30 * 3) + (20 * 3) + (15 * 1) + (10 * 1)$$

50

SCORE = _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

8. What is the distribution of educational backgrounds (highest degrees obtained) of the application staff? Indicate a percentage for each category. (Percentages should total to 100%)

	WEIGHT

_____ Graduate college degree	10
_____ Bachelors college degree	8
_____ Two-year college degree	5
_____ High school diploma or less	1

SCORING

For each percentage, multiply by the corresponding weight. Sum the products and then divide the sum by 100. Maximum score is 10 points, minimum score is 1 point.

Example: For these percentages, the calculations are as follows:

15% graduate college degree
55% bachelors college degree
25% two-year college degree
5% high school diploma or less

$$(15 * 10) + (55 * 8) + (25 * 5) + (5 * 1)$$

100

SCORE = _____

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

9. Overall, in your judgement, to what extent are (or have been) the problems in maintaining the current installed application portfolio? (Check the appropriate category.)

No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem	SCORE
-------------------	------------------------	---------------	------------------------	---------------	-------

a. Turnover of maintenance personnel					
b. Quality of application system documentation					
c. Changes made to application system hardware and software					
d. User demand for enhancements and extensions to application system					
e. Skills of maintenance programming personnel					
f. Quality of original programming of application system					
g. Number of maintenance programming personnel available					
h. Competing demands for maintenance programming personnel time					
i. Inadequate hardware/software configurations in IS Organization					
j. Ability to recruit qualified personnel					
k. Lack of user understanding of application system					
l. Storage requirements of application system programs					
m. Inadequate software tools (debuggers, analyzers, etc..)					
n. Motivation of maintenance programming personnel					

SSQAM -- Supportability Measure
Organization Questionnaire - Scoring Directions

	No Problem At All					
	Somewhat Minor Problem					
	Minor Problem					
	Somewhat Major Problem					
	Major Problem					
	SCORE					
o. Forecasting of maintenance programming personnel requirements						
p. Maintenance programming productivity						
q. Competing demands between new systems development and maintenance						
z. Turnover in user organization						
s. Unrealistic user expectations						
t. Adherence to programming standards in maintenance						
u. Management support of application system						
v. Quality of application system design						
w. Budgetary pressures						
x. Meeting scheduled commitments						
y. Inadequate training of user personnel						

SCORING

- For each lettered item, score
- 5 points for "No problem at all"
 - 4 points for "Somewhat minor problem"
 - 3 points for "Minor problem"
 - 1 points for "Somewhat major problem"
 - 0 point for "Major problem"

Software Supportability Qualitative Assessment Methodology
System Questionnaire Worksheet (Supportability)

NAME OF SYSTEM _____

QUESTIONNAIRE NUMBER _____

Question	System	Process	Resource	Range
1.	_____			0-5
2.	_____			0-5
3.	_____			0-5
4.	_____			0-5
5.				
6.	_____			0-5
7.	_____			0,2
8.	_____			0,2
9.	_____			0,2
10.	_____			0,2
14.	_____			1-4
15.	_____			1-4
19.	_____			0-4
26a.			_____	0-5
b.			_____	0-5
c.	_____			0-5
d.	_____			0-5
e.	_____			0-5
f.	_____			0-5
g.	_____			0-5
h.	_____			0-5
i.	_____			0-5
j.		_____		0-5
o.		_____		0-5
TOTALS	_____			2-80
		_____		0-10
			_____	0-10

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

QUESTIONNAIRE NUMBER _____

Name of Information System: _____

1. What is the size of the system source code, in lines of code (LOC)?

_____ lines of code

SCORING

Calculate the score utilizing the following scale:

System Size			SCORE
At least	But less than		
-----	-----		-----
0	10,000	lines of code	5
10,000	50,000	"	4
50,000	100,000	"	3
100,000	500,000	"	2
500,000	1,000,000	"	1
1,000,000		"	0

SCORE - _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

2. What language(s) is the software written in?

_____	_____
_____	_____
_____	_____

SCORING

Scoring should be based upon the following scale:

Number of languages	SCORE
-----	-----
1	4
2	3
3	2
4	1
Greater than 4	0

Add one additional point to the score if at least half of the languages are high-level, 4th generation languages or later.

Examples of allowable languages

Language	Allowed?
-----	-----
COBOL	No
Assembly	No
C	No
Ada	Yes
DBASE III	Yes
SQL	Yes

SCORE - _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

3. How many modules (units that perform single functions or sets of functions) does the software product contain?

_____ number of modules

SCORING

To calculate a score for this question, you need the answers for both this question and question number one (system size in lines of code).

Calculate the average module size, in lines of code, by dividing the answer in number one by the answer to this question. Then assign a score according to the following scale:

Average Module Size

At Least	But Less Than		SCORE
-----	-----		-----
0	500	lines of code	5
500	1,000	"	4
1,000	2,000	"	3
2,000	3,000	"	2
3,000	5,000	"	1
5,000		"	0

Example: If the information system contains 200,000 lines of code and it contains 300 modules, then the average module size is:

$$200,000 / 300 = 1,250 \text{ lines of code.}$$

Thus, we would assign a score of 3.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

4. What is the age (measured from date of original installation)
of the software product?

_____ age of system (in years)

SCORING

Compute the score using the following scale:

System Age			SCORE
At Least	But Less Than		
0	1	year	5
1	3	years	4
3	6	"	3
6	8	"	2
8	10	"	1
10		"	0

SCORE = _____

5. How long has your organization supported this software product?

_____ length of support (in years)

SCORING

No scoring for this question.

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

6. What are the TOTAL number of changes that have been made to this product (software and associated documentation) during the time you have supported it? Include both Software Change Packages and Emergency Change Packages.

_____ total number of changes

SCORING

To compute the score for this question, you need the answers for both this question and question number 5 (length of support).

Calculate the average number of changes per year by dividing the answer to this question by the answer in number 5. Then assign a score according to the following scale:

Average Number of Changes Per Year			SCORE
At Least	But Less Than		-----
0	5	changes per year	5
5	10	"	4
10	50	"	3
50	100	"	2
100	500	"	1
500		"	0

Example: If the information system has been supported for 5 years, and a total of 175 changes have been implemented to this system, then the average number of changes is:

$$175 / 5 = 35 \text{ changes per year.}$$

Thus, we would assign a score of 3.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

7. Does the software contain any code that aids in debugging the software?

_____ yes SCORE _____
_____ no

8. Is there any documentation explaining the overall function of the software?

_____ yes SCORE _____
_____ no

9. Is there documentation for each module explaining the module's function?

_____ yes SCORE _____
_____ no

10. Are there any user's manuals explaining the use of this software?

_____ yes SCORE _____
_____ no

SCORING

For each of questions 7 through 10, assign a score of 2 points for each "yes" answer and a score of 0 points for each "no" answer.

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

14. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ Small-scale (affect a few lines of code at most)?
_____ Medium-scale (affect several functions or modules)?
_____ Large-scale (affect all or a large portion of the software)?

100 % TOTAL

SCORING

For each percentage, multiply by the corresponding weight
as shown in the following weight table:

Type of Action	WEIGHT
-----	-----
Small-Scale	4
Medium-Scale	3
Large-Scale	1

Sum the resulting products, divide the result by 100,
and round to the nearest integer. Maximum score is 4 points,
minimum score is 1 point.

Example: If the percentage for the various types of
maintenance requests are as follows:

45% Small-scale requests
45% Medium-scale requests
10% Large-scale requests

The calculation is:

$$\frac{(45 * 4) + (45 * 3) + (10 * 1)}{100} = 3.25,$$

which rounds to 3. Thus, the score is 3.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

15. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ EMERGENCY (require immediate attention and must be completed as
soon as possible to ensure the correct operation of the software)

_____ URGENT (require urgent attention -- more so than normal requests --
and must be completed within a relatively short period of time)

_____ NORMAL (require no special attention and can be completed within
the usual framework of support procedures)

100 % TOTAL

SCORING

For each percentage, multiply by the corresponding weight
as shown in the following weight table:

Type of Request	WEIGHT
-----	-----
Emergency	1
Urgent	2
Normal	4

Sum the resulting products, divide the result by 100,
and round to the nearest integer. Maximum score is 4 points,
minimum score is 1 point.

Example: If the percentage for the various types of
maintenance requests are as follows:

10% Emergency
10% Urgent
80% Normal

The calculation is:

$$\frac{(10 * 1) + (10 * 2) + (80 * 4)}{100} = 3.50,$$

which rounds to 4. Thus, the score is 4.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

19. What percentage of time spent maintaining the software is devoted to testing it?

_____ (%) time spent on testing

SCORING

To calculate the score for this question, utilize the following scale:

Percentage Test Time		
At Least	But less Than	SCORE
-----	-----	-----
0%	10%	0
10%	20%	1
20%	30%	2
30%	40%	3
40%		4

SCORE = _____

Software Supportability Qualitative Assessment Methodology
System Questionnaire - Scoring Directions (Supportability)

26. Overall, in your judgment, to what extent are (or have been) the following problems in maintaining this information system?

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem	SCORE
a. Not enough people to support this system.						
b. People supporting this system are not trained adequately.						
c. System is overly large, making support difficult.						
d. System is overly complex, making support difficult.						
e. System is not well-structured (written in "spaghetti code").						
f. Lack of system modularization makes changes difficult to implement.						
g. System is old and needs to be replaced.						
h. System documentation is incomplete or confusing.						
i. System documentation is out-of-date.						
j. Not enough time is spent on testing after changes are made.						
o. User requirements for this system change frequently.						

SCORING

For each lettered item, score

- 5 points for "No problem at all"
- 4 points for "Somewhat minor problem"
- 3 points for "Minor problem"
- 1 point for "Somewhat major problem"
- 0 points for "Major problem"

F Supportability Worksheet - Final Results

This appendix contains a *one-page* worksheet for calculating the final supportability measure.

Software Supportability Qualitative Assessment Methodology
Supportability Worksheet

NAME OF SYSTEM _____

	System -----	Process -----	Resource -----
AVERAGE Organizational Questionnaire Scores	_____	_____	_____
	+	+	+
AVERAGE System Questionnaire Scores	_____	_____	_____
	-	-	-
TOTAL Factor Scores	_____ _____	_____ _____	_____ _____
	<div style="text-align: center;"> \ / \ / \ / + </div>		
ADD Factor Scores	_____		
DIVIDE by 3	----- 3		
FINAL SCORE	_____ _____		

References

- [CDS86] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, CA, 1986.
- [DeM82] T. DeMarco. *Controlling Software Projects*. Yourdon, New York, 1982.
- [Dep82] Department of Defense. *Test and Evaluation of System Reliability, Availability, and Maintainability: a Primer*, March 1982. DoD Directive 3235.1-H.
- [IEE83] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*, 1983. Standard 729-1983.
- [PTH87] D. E. Peercy, E. Tomlin, and G. Horlbeck. Assessing software supportability risk: A minitutorial. In *Conference on Software Maintenance*, pages 72-80. IEEE, 1987.
- [Sch90] Stephen R. Schach. *Software Engineering*. Aksen Associates Incorporated Publishers, Boston, MA, 1990.
- [Som89] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, New York, 1989.
- [War87] J. L. Warthman. Software quality measurement demonstration project (i). Technical Report RADC-TR-87-247, Rome Air Development Center, December 1987.